NASA/TM–2001-210929

Improving Simulated Annealing by Recasting it as a Non-Cooperative Game

David H. Wolpert, Esfandiar Bandari, and Kagan Tumer

September 2001

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM–2001-210929

Improving Simulated Annealing by Recasting it as a Non-Cooperative Game

David Wolpert
Ames Research Center, Moffett Field, California

Esfandiar Bandari
RIACS, Ames Research Center, Moffett Field, California

Kagan Tumer
Ames Research Center, Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

September 2001

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

Improving Simulated Annealing by Recasting it as a Non-Cooperative Game

David H. Wolpert
NASA Ames Research Center
dhw@ptolemy.arc.nasa.gov

Esfandiar Bandari
RIACS, NASA Ames Research Center
bandari@mail.arc.nasa.gov

Kagan Tumer
NASA Ames Research Center
kagan@ptolemy.arc.nasa.gov

Abstract

The game-theoretic field of COllective INtelligence (COIN) concerns the design of computer-based players engaged in a non-cooperative game so that as those players pursue their self-interests, a pre-specified global goal for the collective computational system is achieved "as a side-effect". Previous implementations of COIN algorithms have outperformed conventional techniques by up to several orders of magnitude, on domains ranging from telecommunications control to optimization in congestion problems. Recent mathematical developments have revealed that these previously developed game-theory-motivated algorithms were based on only two of the three factors determining performance. Consideration of only the third factor would instead lead to conventional optimization techniques like simulated annealing that have little to do with non-cooperative games. In this paper we present an algorithm based on all three terms at once. This algorithm can be viewed as a way to modify simulated annealing by recasting it as a non-cooperative game, with each variable replaced by a player. This recasting allows us to leverage the intelligent behavior of the individual players to substantially improve the exploration step of the simulated annealing. Experiments are presented demonstrating that this recasting improves simulated annealing by several orders of magnitude for spin glass relaxation and bin-packing.

1 INTRODUCTION

There are two general types of distributed systems that are found in nature and that researchers have translated into computational algorithms for function maximization. The first is exemplified by Neo-Darwinian natural selection, which has been translated into genetic algorithms (GA's) [5]. The function G maximized in these distributed systems takes as argument any single one of the system's variables. (Each of those variables is viewed as a "genome", with G of a genome being the "fitness" of the "phenotype" induced by that genome.)

Whereas systems of this first type have a "narrow G", in the second type of distributed system the function G being optimized is "wide", taking the state of the entire distributed system as its argument. In some such distributed systems it is only in the crudest sense that the individual variables can be viewed as players in a non-cooperative game. Examples include simulated annealing (SA [13]) and swarm intelligence [1], inspired by spin relaxation in physics and eusocial insect colonies, respectively.

In other distributed systems that have a wide G the value of each of the individual variables going into G is set by a player engaged in an over-arching non-cooperative game where each player η is trying to maximize its associated payoff utility function g_η. Roughly speaking, such collective systems work when the utility functions of the individual variables/players are all "aligned" with the world utility G. Under these circumstances, as the individual players pursue their self-interests, the global goal for the full collective of maximizing G is achieved "as a side-effect". The primary naturally-occurring analogues of such collectives are economic institutions where the players are human beings, e.g., auctions and clearing of markets. In the computational translations of such systems the players are computer programs [20, 4, 12].

The "COllective INtelligence" (COIN) framework concerns the design of such collectives. In particular, it addresses the issue of how to generate from a provided G the set of utilities $\{g_\eta\}$ that have optimal signal/noise for each player η while also having the property that as the individual players maximize those utilities, G also gets maximized. This work on design of collectives can be viewed as an extension of mechanism design [9] beyond human economics, to include concern for signal-to-noise ratio in the payoff functions and off-equilibrium behavior, and to allow far more freedom in choice of the g_η than exists with human players (for example to encompass computational systems in which the issue of incentive compatibility is moot), and also to encompass arbitrary G and arbitrary dynamics of the system. Applications of this framework on problems from routing in telecommunication networks [21, 24] to congestion problems [25] have resulted in substantial performance improvement over conventional techniques that do not consider issues like signal-to-noise. Typically as the size of the collective grows, such improvements reach several orders of magnitude.

Recent mathematical developments have shown that the previously developed COIN algorithms for design of collectives were based on only two of the three factors determining performance at maximizing G. Consideration of only the third factor would instead lead to conventional wide-G systems like simulated annealing that have little to do with non-cooperative games. Consideration of all three terms at once therefore would result in an algorithm that combines the two types of wide-G function maximization system, with naturally-occurring analogues of human economics and statistical physics, respectively.

In this paper we present such an algorithm. Because of its similarity to (certain aspects of) how human corporations are run, we call it the Computational Corporation (CoCo) algorithm. Roughly speaking, it works by modifying the exploration step of simulated annealing by having the new values of the variables set by the moves of intelligent players in a non-cooperative game. Like simulated annealing, the computational corporation algorithm is intended not to give the best possible performance in all problem domains — an algorithm laboriously tailored for a particular domain will invariably perform best for that domain [23]. Rather like other algorithms related to naturally-occurring distribution systems, the computational corporation algorithm is intended as a powerful and broadly applicable "off-the-shelf" algorithm.

We present experiments demonstrating that the computational corporation algorithm outperforms simulated annealing by several orders of magnitudes for spin glass relaxation and bin-packing. In the spin glass domain CoCo converges to a given value of G over two orders of magnitude faster than does SA, with far better scaling behavior (the ratio of their convergence speeds increased exponentially with the size of the problem). In the bin packing problem, both CoCo and conventional COIN algorithms significantly outperform SA (up to three orders of magnitude faster convergence). CoCo achieves the optimum solution in a higher percentage of the runs (82 % vs. 56 %) than does the COIN algorithm, and provides better "worse case" properties (i.e., the worst result obtained through CoCo is better than the worst result obtained through COIN).

2 The Mathematics of Collective Intelligence

The full formalization of the COIN framework extends significantly beyond what is needed for this paper.[1] The restricted version needed here starts with an arbitrary vector space Z whose elements ζ give the state of all the variables in the collective.

We wish to search for the ζ that maximizes the provided **world utility** G. In addition to G we are concerned with **payoff utility functions** $\{g_\eta\}$, one such function for each variable/player η. We use the notation $\hat{\eta}$ to refer to all players other than η.

We will need to have a way to "standardize" utility functions so that the numeric value they assign to a ζ only reflects their ranking of ζ relative to certain other elements of Z. We call such a standardization of utility U for player η the "**intelligence** for η at ζ with respect to U". Here we will use intelligences that are equivalent to percentiles:

$$\epsilon_U(\zeta : \eta) \equiv \int d\mu_{\zeta_{\hat{\eta}}}(\zeta')\Theta[U(\zeta) - U(\zeta')] \,, \tag{1}$$

where the subscript on the (normalized) measure indicates it is restricted to ζ' sharing the same non-η components as ζ, and where the Heaviside function Θ is defined to equal 1 when its argument is greater than or equal to 0, and to equal 0 otherwise. Note that an intelligence value is always between 0 and 1.

Our uncertainty concerning the behavior of the system is reflected in a probability distribution over Z. Our ability to control the system consists of setting the value of some characteristic of the collective, e.g., setting the payoff functions of the players. Indicating that value by s, our analysis revolves around the following central equation for $P(G \mid s)$, which follows from Bayes' theorem:

$$P(G \mid s) = \int d\epsilon_G P(G \mid \epsilon_G, s) \int d\epsilon_g P(\epsilon_G \mid \epsilon_g, s) P(\epsilon_g \mid s) \,, \tag{2}$$

where "ϵ_g" is the vector of the intelligences of the players with respect to their associated payoff functions, and "ϵ_G" is the vector of the intelligences of the players with respect to G.

If we can choose s so that the third term in the integrand is peaked around vectors ϵ_g all of whose components are close to 1, then we have likely induced large (payoff function) intelligences. If we can also have the second term be peaked about ϵ_G equal to ϵ_g, then ϵ_G will also be large. Finally, if the first term in the integrand is peaked about high G when ϵ_G is large, then our choice of s will likely result in high G, as desired.

Intuitively, it is in the third term that the requirement that payoff functions have high "signal-to-noise" (an issue not considered in conventional work in mechanism design) arises. It is in the second term that the requirement that the payoff functions be "aligned with G" arises. Previously developed COIN algorithms concentrated on these two terms. Finally, conventional function maximization techniques like simulated annealing instead are concerned with having term 1 have the desired form.

It is the simultaneous concern for all three of these terms that underlies the CoCo algorithm. To present that algorithm we must first review some COIN results on how to simultaneously set terms 2 and 3 to have the desired form.

The second term in Eq. 2 can be addressed by requiring that the collective be **factored**, which means that ϵ_g equals ϵ_G exactly for all ζ. In game-theory language, the Nash equilibria of

[1] That framework encompasses, for example, arbitrary reassignments of how the various subsets of the variables comprising the collective are assigned to players. It also encompasses modification of the players' information sets (i.e., modification of inter-player communication). See [22].

a factored collective are critical points of G. In addition to this desirable equilibrium behavior, factored collectives incorporate appropriate off-equilibrium incentives to the players. As a trivial example, any "team game" in which all the payoff functions equal G is factored [8]. However team games often have very poor forms for term 3 in Eq. 2, forms which get progressively worse as the size of the collective grows. This is because for such payoff functions player each η will usually confront a very poor "signal-to-noise" ratio in trying to discern how its actions affect its payoff $g_\eta = G$, since so many other players' actions affect G and therefore affect g_η.

Previous COIN algorithms were based on varying the payoff functions $\{g_\eta\}$ to optimize the third term, subject to the requirement that the system be factored. To understand how those algorithms work, given a measure $d\mu(\zeta_\eta)$, define the **opacity** at ζ of utility U as:

$$\Omega_U(\zeta : \eta, s) \equiv \int d\zeta' J(\zeta' \mid \zeta) \frac{|U(\zeta) - U(\zeta'_{\hat\eta}, \zeta_\eta)|}{|U(\zeta) - U(\zeta_{\hat\eta}, \zeta'_\eta)|} , \tag{3}$$

where J is defined in terms of the underlying probability distributions.[2]

The denominator absolute value in the integrand in Eq. 3 reflects how sensitive $U(\zeta)$ is to changing ζ_η. In contrast, the numerator absolute value reflects how sensitive $U(\zeta)$ is to changing $\zeta_{\hat\eta}$. So the smaller the opacity of a payoff function g_η, the more $g_\eta(\zeta)$ depends only on the move of player η, i.e., the better the associated signal-to-noise ratio for η. Intuitively then, lower opacity means it is easier for η to achieve a large value of its intelligence.

To formalize this, choose a measure $d\mu$ defining opacity that is the same as the one defining intelligence. Then expected opacity bounds how close to 1 expected intelligence can be [22]:

$$\begin{aligned} &E(\epsilon_U(\zeta : \eta) \mid s) \leq 1 - K, \text{where} \\ &K \leq E(\Omega_U(\zeta : \eta, s) \mid s). \end{aligned} \tag{4}$$

In practice the bounds in this result are usually tight.

Next define a **difference** utility as one of the form:

$$U(\zeta) = G(\zeta) - \Gamma(f(\zeta)) \tag{5}$$

where $\Gamma(f)$ is independent ζ_η. In general it is not possible for a collective both to be factored and to have zero opacity for all of its players. However any difference utility is factored [22]. In addition, under usually benign approximations, $E(\Omega_u \mid s)$ is minimized over the set of difference utilities by choosing:

$$\Gamma(f(\zeta)) = E(G \mid \zeta_{\hat\eta}, s) , \tag{6}$$

up to an overall additive constant. We call the resultant difference utility the **Aristocrat** utility (AU), loosely reflecting the fact that it measures the difference between a player's actual action and the average action.

If possible, we would like each player η to use the associated AU as its payoff function. This is not always feasible however. The problem is that to evaluate the expectation value defining its AU each player needs to evaluate the current probabilities of each of its potential actions. However if the player then changes its payoff function to be the associated AU it will in general substantially change its ensuing behavior those probabilities. (The player now wants to choose

[2] Writing it out in full, $J(\zeta' \mid \zeta) \equiv J(\zeta_\eta, \zeta' \mid \zeta_{\hat\eta}, s)/P(\zeta_\eta \mid \zeta_{\hat\eta}, s)$, with:

$$J(\zeta_\eta, \zeta' \mid \zeta_{\hat\eta}, s) \equiv \frac{P(\zeta_\eta \mid \zeta_{\hat\eta}, s)P(\zeta'_\eta \mid \zeta_\eta, s)\mu(\zeta'_\eta)}{2} + \frac{P(\zeta'_\eta \mid \zeta'_{\hat\eta}, s)P(\zeta_\eta \mid \zeta'_\eta, s)\mu(\zeta_\eta)}{2} .$$

actions that maximize a different function from the one it was maximizing before.) In other words, it will change the probabilities of its actions, which means that its new payoff function is in fact not the AU for its actual probabilities. There are ways around this self-consistency problem, but in practice it is often easier to bypass the entire issue, by giving each η a payoff function that does not depend on the probabilities of η's own actions.

One such payoff function is the **Wonderful Life** Utility (WLU). The WLU for player η is parameterized by a pre-fixed **clamping element** CL_η chosen from among η's possible actions:

$$WLU_\eta \equiv G(\zeta) - G(\zeta_{\hat{\eta}}, CL_\eta) . \tag{7}$$

WLU is factored no matter what the choice of clamping element. Furthermore, while not matching the low opacity of AU, WLU usually has far better opacity than does a team game.

In many circumstances one can meaningfully interpret a particular choice of clamping element for player η as equivalent to a "null" action for player η, equivalent to removing that player from the system. (Hence the name of this payoff function — cf. the Frank Capra movie.) For such a clamping element assigning the associated WLU to η as its payoff function is closely related to the economics technique of "endogenizing a player's externalities" [9]. However it is usually the case that using WLU with a clamping element that is as close as possible to the expected action defining AU results in far lower opacity than does clamping to the null action. Accordingly, use of such an alternative WLU almost always results in far better values of G than does the "endogenizing" WLU.

Typically, COINs in which the payoff functions are WLU or AU not only far outperform team games, but also conventional function maximization techniques like simulated annealing. However note that even if the payoff functions result in the collective's having every component of the vector ϵ_G equal 1 — the best terms 2 and 3 can be — nothing in Eq. 2 precludes a poor value for $G(\zeta)$. This is because having all those intelligences equal 1 only means that the collective is at a *local* maximum of G.

This potential shortcoming is reflected in the first term in Eq. 2, a term that does not directly depend on the choice of the players' payoff functions. Crudely speaking, what that term reflects is the propensity of the system to get stuck in a local maximum. Accordingly, one can use many of the conventional exploration/exploitation function maximization techniques like simulated annealing to induce a good form for that term. At each iteration, the exploration step is determined by the actions chosen by the players, rather than by using one of the more "blind" sampling schemes that are traditionally employed. The exploitation step though is the same as in the traditional formulation of the algorithm. In this way all three terms of Eq. 2 will have a desired form, and the induced G should be large.

In its concern for all three terms this algorithm bears many similarities to well-run modern human corporations, with G the "bottom line" of the entire corporation, the players η identified with the employees of the corporation, and the associated g_η being the employees' performance-based compensation packages. For example, for a "factored corporation", each employee's compensation package function contains incentives designed so that the better the employee performs their job, the better the bottom line of the corporation. In addition, if the compensation packages are "low opacity", the employees will have a relatively easy time discerning the relationship between their behavior and their compensation. Finally, the centralized exploitation process in CoCo is similar to the centralized decision-making of upper management that tries to determine whether to abandon or stick with a particular set of behaviors by the employees. It is due to these similarities that we call this algorithm the computational corporation algorithm.

3 Experiments

The purpose of this section is to show both the wide applicability of CoCo, and to provide a comparative analysis showing its superiority over Simulated Annealing. For this purpose we chose two diverse domains of applications:

- Minimum energy configurations for binary spin glasses (Theoretical physics)
- Bin packing (Industrial engineering and resource allocation)

In the experiments reported below, for simulated annealing, each agent (spin, item) had a 25% probability of changing its actions (i.e., on the average, the new state differed by 25% from the previous state). The annealing schedule consisted of reducing the temperature (multiply by 0.9) after a fixed number of time steps had elapsed (500 for spin glasses (on convergence runs reported in the text), 100 for bin packing). The players in the CoCo algorithm were handicapped by using perhaps the simplest possible reinforcement learning algorithm [6, 18, 24, 25]. The AU version of CoCo simply assumed that each agent η had a uniform probability distribution over its possible moves. Unless otherwise specified, the clamping elements in the COIN versions were set to $\vec{0}$ (vector of zeroes).

3.1 Binary Spin Glasses — The 2-D Ising Model

Spin glasses have traditionally been viewed as one of the pillars of statistical mechanics and the preferred comparative domain for analyzing the efficacy of various stochastic relaxation and optimization techniques [14]. There are many optimization methods developed precisely for solving the spin glass problem [14] and in this article we are not aiming to improve upon such specialized methods. Rather, we use this domain to compare two multi-purpose optimization algorithms, namely CoCo and simulated annealing.

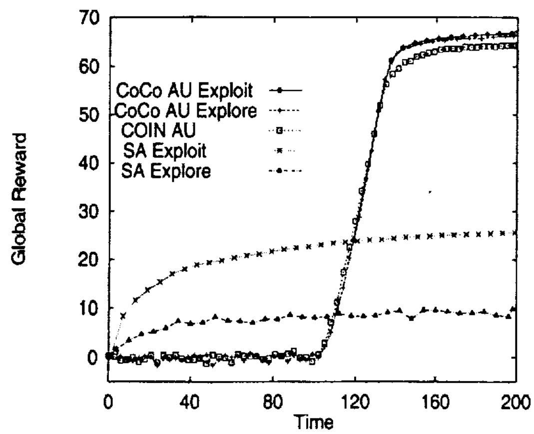
In this article, we restrict our attention to the 2-D Ising Model, i.e., a special spin glass where each site can occupy only one of two possible states - spin up or down in ferromagnetism, and empty or occupied in modeling liquid/gas phase transition. Because exact algorithmic solutions for the two dimensional Ising model have been already developed, 2-D binary spin glasses are considered a standard benchmarking tool [14]. Briefly, this problem consists of (a four connected) two dimensional grid with periodic boundary conditions. Each site s_k can have one of two spins (here taken to be 1 or -1). The link between any two sites, $l_{i,j}$, is an arbitrary value (here, without loss of generality restricted to between -1 and 1). The goal is to find the states of the spin glass such that the global energy defined by:

$$\sum_i \sum_j s_i . s_j . l_{i,j}$$

is minimized. It is easy to show that this problem has many local minima, and for any reasonable sized grid, examining all possible states quickly becomes an intractable problem [14] (2^n states for n sites).

In modeling this binary grid, we mapped each site ζ_k as an independently active agent, with its' chosen action at time t represented by the binary choice $s_{k,t}$ (i.e., the choice of spin up or spin down). Each agent selects its next action/state using the COIN/CoCo framework.

Figure 1(a) shows the performance (averaged over 50 runs) of Simulated Annealing vs. AU CoCo and AU COIN for a 10x10 Ising model. As it can be seen, CoCo far outperforms Simulated Annealing. Comparing the exploration steps of CoCo and simulated annealing shows the advantage gained through CoCo. Notice that the gap between the exploration and exploitation steps



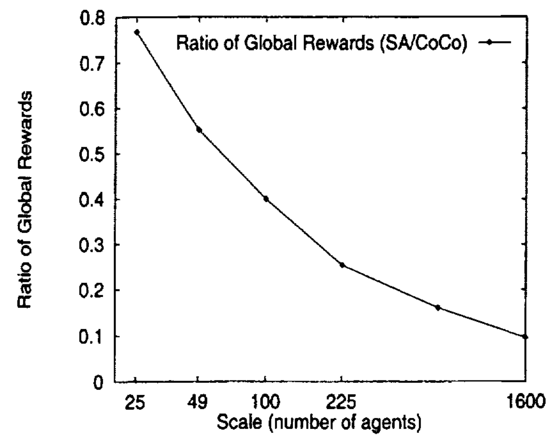


Figure 1: (a) Performance (left); (b) Scaling (right) for the spin glass problem.

of CoCo are far narrower than that of simulated annealing, emphasizing the "guided" aspect of the search used by CoCo.

Comparing the performance of SA to that of CoCo, one interesting question that arises is whether SA will reach the CoCo results given sufficient time, and if so, how much longer will it take to do so. On the 10x10 grid discussed above, the best simulated annealing schedule used required two orders of magnitude longer to convergence, and even then failed to reach the CoCo results[3]. As the scaling runs discussed below indicate, one expects this discrepancy to increase with the size of the problem.

A final noteworthy result is the increased scalability obtained through CoCo. Figure 1(b) shows the ratio of the the global utility achieved using simulated annealing to the global utility achieved through CoCo as a function of the grid size. The exponential decline in performance clearly shows that the larger the size of the binary spin glass grid, the greater the impact of CoCo.

3.2 Bin-Packing

The second problem we investigate is the bin packing problem [7]. This problem consists of a list of items $(a_1, a_2, \cdots, a_n)$ and a supply of n bins, B_i, with capacity C each. The size of each item is given by a function $s(a_i)$ which satisfies $0 \leq s(a_i) \leq C, \forall i$. The problem consists of packing the items into a minimum number of bins, while ensuring that the contents of no bin exceeds C. More precisely, the problem consists of minimizing:

$$G = \sum_{i=1}^{n} I_{B_i}, \qquad \text{subject to:} \qquad \sum_{i \in B_i} s(a_i) \leq C \tag{8}$$

where I_{B_i} is the "content-indicator" function for B_i, and is 0 if the bin is empty and 1 otherwise.

This problem has many real world application ranging from loading trucks subject to weight limitations [7] to distributing jobs on a computer network where each processor has limited resources (e.g., memory, CPU cycles) [16]. The bin-packing problem is known to be NP-complete [10], and many approximate algorithms were developed to address it [3, 7]. In this section we study how the COIN framework can be applied to this domain.

[3]An optimistic extension of the SA performance under ideal annealing schedules projects SA reaching CoCo results 500 times more slowly than CoCo in this relatively small problem.

In these experiments the agents use a "soft" version of the global utility function given by:

$$G_{\text{soft}} = \begin{cases} \sum_{i=1}^{n} \left[\left(\frac{C}{2}\right)^2 - \left(x_i - \frac{C}{2}\right)^2 \right] & \text{if } x_i \leq C \\ \sum_{i=1}^{n} \left(x_i - \frac{C}{2}\right)^2 & \text{if } x_i > C \end{cases} , \tag{9}$$

where $x_i = \sum_{i \in B_i} s(a_i)$ gives the total size of all items in bin i. This function has two minima (at 0 and C) and provides two benefits: First, by discouraging "illegal" solutions due to the large penalty incurred by exceeding C in any one bin, it greatly reduces the need to verify that a solution satisfies the constraints *after* that solution is found. Second, it provides the system with a better "signal" and encourages bins to be closer to full or empty. All algorithms (including simulated annealing) use this soft G function, but they are all evaluated based on the provided global reward (Eq 8).

In these experiments, all the algorithms had the same number of iterations (1000 in this case) and the results we report below averaged over 50 runs[4]. Note that COIN-based systems used the first 200 steps to generate their "learning" data, and thus took random actions during this interval. In simulated annealing, the proposal distribution was slowly modified to generate solutions that differed in fewer items than the current solution as the experiment progressed. The annealing schedule consisted of reducing the Boltzmann temperature at intervals of 100 steps.

Table 1: Performance at $t = 1000$

Algorithms	Average	Best	Worst	Reached Optimal
CoCo WLU	4.17 ± 0.05	4	5	82 %
CoCo TG	14.28 ± 0.16	12	16	0 %
COIN WLU	4.46 ± 0.08	4	6	56 %
COIN TG	15.76 ± 0.23	13	18	0 %
Sim Anneal	15.67 ± 0.18	12	18	0 %

Table 1 summarizes the results of the various algorithms for the bin packing problem. The average performance of simulated annealing and team game COIN were statistically indistinguishable. Neither fared well, with the worst solution in both cases being random. CoCo team game performed slightly better, but the real gains were not achieved until the WLU private utility function was used[5]. Though the CoCo WLU slightly outperformed the straight COIN WLU (lower average and higher percentage of finding the optimal solution) both of these algorithms were significantly superior to the other three.

Although the results reported above show the superiority of the WLU-based algorithms, they do not fully reflect the advantages of COIN-based systems. One aspect of the algorithm performance that is of paramount importance in optimization problems is the speed of convergence. The two WLU-based algorithms both converged to near optimal solutions within the first 50 steps following the learning period. Even projecting the team game CoCo and simulated annealing performances linearly[6] they were two and three orders of magnitude slower, respectively, than WLU-based algorithms.

[4]The errors in the mean are reported as plus/minus in the table, and omitted in the graph because the resulting error bars are too small to see.

[5]For this problem, there was no difference (statistically) between the performance of WLU and AU. Therefore to streamline the comparative process, we report only WLU results.

[6]This favors TG and SA since in reality their convergence rate drops.

4 Conclusion

There are three general types of parallel systems found in nature that can be viewed as engaging in maximization of a function G. These are exemplified by neo-Darwinian natural selection (for G that take any single one of the elements of the parallel system as an argument), spin glass relaxation (for G that take the entire system as argument), and clearning of markets in economics relaxation (for G that take the entire system as argument and in which the overall parallel system can be viewed as a non-cooperative game). All three types of system have been translated into computational algorithms, exemplified by genetic algorithms, simulated annealing, and computational markets, respectively.

The Collective Intelligence framework can be viewed as an extension of conventional economics-based systems of the third type, to reflect signal-to-noise issues and greater freedom in modifying the individual players than exist in economies of human beings. It has traditionally been applied only to systems of the third type. Recent mathematical advances in that framework have shown that those traditional COIN algorithms only account for two of the three factors determining performance. The third factor can be accounted for by integrating the COIN with a technique of the second type, like simulated annealing. Intuitively, such an integrated system, which we call a computational corporation, can be viewed as conventional simulated annealing modified by having the value of each variable in the exploration step of the SA be set by a (computer-based) player in an associated non-cooperative game. Doing this allows the leveraging of the intelligence of such players to improve the exploration, and thereby improve the performance.

We present experiments demonstrating that the computational corporation algorithm outperforms simulated annealing by several orders of magnitudes for spin glass relaxation and bin-packing. In the spin glass domain CoCo converges to a given value of G over two orders of magnitude faster than does SA, with far better scaling behavior (the ratio of their convergence speeds increased exponentially with the size of the problem). In the bin packing problem, both CoCo and conventional COIN algorithms significantly outperform SA (up to three orders of magnitude faster convergence).

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from social insect behaviour. *Nature*, 406(6791):39–42, 2000.

[2] C. Boutilier, Y. Shoham, and M. P. Wellman. Editorial: Economic principles of multi-agent systems. *Artificial Intelligence Journal*, 94:1–6, 1997.

[3] J. A. Boyan and A. Moore. Learning evaluation functions for global optimization and boolean satisfiability. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. AAAI Press, 1998.

[4] D. Challet and Y. C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514, 1998.

[5] K. Chellapilla and D.B. Fogel. Evolution, neural networks, games, and intelligence. *Proceedings of the IEEE*, pages 1471–1496, September 1999.

[6] C. Claus and C. Boutilier. The dynamics of reinforcement learning cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, Madison, WI, June 1998.

[7] E. G. Coffman Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In *Approximation Algorithms for NP-Hard Problems*, pages 46–93. PWS Publishing, 1996.

[8] R. H. Crites and A. G. Barto. Improving elevator performance using reinforcement learning. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 1017–1023. MIT Press, 1996.

[9] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, Cambridge, MA, 1991.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, San Fransisco, 1979.

[11] J. Hu and M. P. Wellman. Online learning about other agents in a dynamic multiagent system. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 239–246, May 1998.

[12] B. A. Hubermann and T. Hogg. The behavior of computational ecologies. In *The Ecology of Computation*, pages 77–115. North-Holland, 1988.

[13] S. Kirkpatrick, C. D. Jr Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[14] B. M. McCoy and T. T. Wu. *The Two-Dimensional Ising Model.* Harvard University Press, Boston, MA, 1973.

[15] T. Sandholm and V. R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94:99–137, 1997.

[16] B. A. Shirazi, A. R. Hurson, and K. M. Kavi. *Scheduling and Load Balancing in Parallel and Distributed Systems.* IEEE Computer Society Press, 1995.

[17] Y. Shoham and K. Tanaka. A dynamic theory of incentives in multi-agent systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.

[18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[19] L. Tesfatsion. How economists can get ALIFE. In Arthur *et al.*, editor, *The Economy as an Evolving System*, volume 27. Adison Wesley, 1997.

[20] P. Tucker and F. Berman. On market mechanisms as a sofware techniques. Technical Report CS96-513, University of California, San Diego, December 1996.

[21] K. Tumer and D. H. Wolpert. Collective intelligence and Braess' paradox. In *Proceedings of the Seventeeth National Conference on Artificial Intelligence*, pages 104–109, 2000.

[22] D. H. Wolpert. The mathematics of collective intelligence. pre-print, 2001.

[23] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. Best Paper Award.

[24] D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press, 1999.

[25] D. H. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6), March 2000.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 2001

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Improving Simulated Annealing by Recasting It as a Non-Cooperative Game

5. FUNDING NUMBERS
755-07

6. AUTHOR(S)
David Wolpert, Esfandiar Bandari, Kagan Tumer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM–2001-210929

11. SUPPLEMENTARY NOTES
Point of Contact: Author, Ames Research Center, MS 269-1, Moffett Field, CA 94035-1000
(650) 604-3362

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited
Subject Category 59 Distribution: Standard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The game-theoretic field of COllective INtelligence (COIN) concerns the design of computer-based players engaged in a non-cooperative game so that as those players pursue their self-interests, a pre-specified global goal for the collective computational system is achieved "as a side-effect". Previous implementations of COIN algorithms have outperformed conventional techniques by up to several orders of magnitude, on domains ranging from telecommunications control to optimization in congestion problems. Recent mathematical developments have revealed that these previously developed game-theory-motivated algorithms were based on o nly two of the three factors determining performance. Consideration of only the third factor would instead lead to conventional optimization techniques like simulated annealing that have little to do with non-cooperative games. In this paper we present an algorithm based on all three terms at once. This algorithm can be viewed as a way to modify simulated annealing by recasting it as a non-cooperative game, with each variable replaced by a player. This recasting allows us to leverage the intelligent behavior of the individual players to substantially improve the exploration step of the simulated annealing. Experiments are presented demonstrating that this recasting improves simulated annealing by several orders of magnitude for spin glass relaxation and bin-packing.

14. SUBJECT TERMS
collective intelligence, algorithm, optimization techniques, simulated annealing, game theory,

15. NUMBER OF PAGES
~~13~~ 14

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		